

**I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.**

# CS 1316 Exam 1

## Summer 2009

Section/Problem	Points Earned	Points Possible
1. Vocabulary Matching		32
2. Fill in the Blank		12
3. Multiple Choice		4
4. Code Understanding		9
5. Turtle Graphics		6
6. Write Code: SumTo		12
7. Better Dorm		15
8. Convert Picture		10
Total Points:		100

## 1. Vocabulary Matching (32 points)

Write the number before the definition on the right on the line before the matching vocabulary word.

___ Array	1. A detailed description of structure and behavior.
___ Block	2. Execution of a model.
___ Boolean	3. A data structure that uses nodes/points and their edges/connections to model aspects of the real world.
___ Boolean Expression	4. A data structure made up of a collection of nodes, where each node points to the next node in line.
___ Cast / Casting	5. The earliest object oriented programming language.
___ Class	6. An early object oriented programming language created by Alan Kay.
___ Constructor	7. The process of telling the compiler what type of data a variable will hold.
___ Data Encapsulation	8. A data type that holds numbers with fractional components.
___ Double	9. A data type that holds numbers without a fractional component.
___ Field	10. A data type that holds sequences of characters.
___ Final	11. A data type that has only two possible values.
___ Graph	12. A section of code, typically enclosed with curly brackets {} that makes up the body of a loop, function, or conditional.
___ Inheritance	13. A logical statement that evaluates to either True or False.
___ Int	14. The fundamental building block of Java programs, they act as the blueprints from which objects are constructed, including definitions of fields and methods.
___ Iterate	15. The instantiation of a class, they have fields that store state and methods (functions) that encode behavior.
___ Linked List	16. The process of hiding internal state from direct access by outsiders, and instead requiring that all accesses to the internal state be done through methods that can act as gatekeepers.
___ Method	17. A keyword that means that anybody can see and manipulate a particular field or method.
___ Method Signature	18. A keyword that means that only methods within the object can access a particular field or method.
___ Model	19. A keyword that means that a field will not change.
___ Object	20. A keyword that means a field or method belongs to the class, and not specific objects that are instantiated from it.
___ Private	21. A method that is called when a new instance is created.
___ Public	22. A class that inherits behaviors and state (methods and fields) from a superclass.
___ Simula	23. The process of extending a superclass, gaining it's behaviors and state.
___ Simulation	24. A class that is extended by subclasses.
___ Smalltalk	25. A keyword that indicates "nothing", as in returns nothing.
___ Static	26. The process of forcing Java to convert data from one data type to another.
___ String	27. These contain state within an object, sometimes called object variables.
___ Subclass	28. A function that is associated with an object and implements behavior.
___ Superclass	29. A homogeneous linear collection of objects which are stored together in memory.
___ Traverse	30. The unique collection of method name and parameter number and type that define a method. Multiple methods may share the same name as long as the type or number of their parameters differ.
___ Type Declaration	31. To (potentially) repeat the execution a block of code multiple times, as with a for or while loop.
___ Void	32. To move through a linear data structure (or a sequence) doing something to or with each individual element.

## 2. Fill in the Blank ( 12 points)

In Java, a \_\_\_\_\_ is used to indicate assignment, while a \_\_\_\_\_ is used for equality checking.

Most statements in Java must be ended with a \_\_\_\_\_. The only exception is if you have only a single statement alone in a \_\_\_\_\_.

In Java, logical **and** is written using the \_\_\_\_\_ symbol, and logical **or** is written using the \_\_\_\_\_ symbol.

Assume that the Student class is a subclass of the Person class, and the Person class is a subclass of the Human class. A variable that is defined to be of type Person can refer to (hold) an object of type \_\_\_\_\_ or type \_\_\_\_\_ but a variable defined to be of type Student can only refer to an object of type \_\_\_\_\_.

A picture that is 200 pixels wide and 100 pixels high has a total of \_\_\_\_\_ pixels. Each pixel needs \_\_\_\_\_ BYTES, to represent the color of the pixel. How many total BITS are used to represent the 200 by 100 picture? Answer: \_\_\_\_\_

## 3. Multiple Choice (4 points)

Circle the correct answer.

**Which of the following is an incorrect conditional statement?**

<p>A.</p> <pre>if ( thisColor == myColor ) setColor( thisPixel , newColor );</pre>	<p>B.</p> <pre>if ( thisColor == myColor ) { setColor( thisPixel , newColor ); }</pre>
<p>C.</p> <pre>if ( thisColor == myColor ) {x = 12; setColor ( thisPixel , newColor ) ; } ;</pre>	<p>D. All are correct.</p>

**The new operator:**

- A. invokes the constructor of an object.
- B. instantiates a specific instance of a class.
- C. allocates memory space for the object.
- D. A and C only.
- E. A, B, and C.

#### 4. Code Understanding ( 9 points)

What does the following code output?

```
int [] a = new int[10];
for ( int i = 0; i < 10; i++) {
    a[i] = 9 - i;
}

for( int i = 0; i < 10; i++) {
    int index = a[i];
    a[i] = a[ index ];
}

for(int i = 0; i < 10; i++) {
    System.out.println( a[i] );
}
```

#### 5. Turtle Graphics ( 6 points)

The following code creates a turtle and moves it around. Draw the turtle's path in the box on the right:

```
Turtle t = new Turtle(new World());

for( int i = 0; i < 20; i++) {

    if (i %2 == 0) {
        t.forward( 10 * i );
    } else {
        t.turn(90);
    } // end if/else

} // end for
```

## 6. Write Code (12 points)

Write (twice) a method `sumTo` which returns the sum of the first  $n$  reciprocals. In other words, `sumTo( n )` returns:

$$1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + \dots + 1/n$$

You may assume that  $n$  is never a negative number. `sumTo(0)` returns 0.0, and `sumTo(1)` returns 1.0. *The first time you write the method, use a for or while loop. The second time you write the method, you must use recursion instead of a loop.* The return value and header for the method(s) are the same:

```
// sumTo with a for or while loop
public static double sumTo(int n) {
```

```
}
```

```
// sumTo using recursion
public static double sumTo(int n) {
```

```
}
```

## 7. Create an Object: Better Dorm (15 points)

Examine the source code for the “Dorm” object provided at the end of this exam.

Write code for a subclass of the Dorm object called **BetterDorm**. Your subclass must have:

- A constructor that accepts a string and two int's representing the name of the dorm and the number of men and women in the dorm.
- A method called **percentMale** that takes no parameters and returns the percentage of the dorm's population that is male (as a double). Your function must work for all possible numbers of men & women in the dorm.

## 8. Convert a picture (10 points)

Write a new method for the Picture class that will return a black and white copy of itself. The method should make a new copy of itself, convert the pixels to B&W (using the following formula), and return it. Your method must be named **bwCopy**, take no parameters, and return a Picture object.

Use the following formula to convert from R,G,B values to a single Luminance (Y) value to put into all 3 (r,g,b) color slots :

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

**Definition of the “DORM” class:**

```
public class Dorm {

    private String myName;
    private int myNumMen;
    private int myNumWomen;

    // Constructor
    public Dorm( String name, int men, int women) {
        myName = name;
        myNumMen = men;
        myNumWomen = women;
    }

    public void setNumMen( int men)
    { myNumMen = men; }

    public void setNumWomen( int women)
    { myNumWomen = women; }

    public void setName( String name)
    { myName = name;}

    public int getNumMen()
    {    return myNumMen; }

    public int getNumWomen()
    {    return myNumWomen; }

    public String getName()
    {    return myName; }

} // end dorm
```